DEMO 1:

CLUSTER INFORMATION :-

ONOS instances (username:onos, password: root)	Ip address
ONOS instance 1	172.19.37.52
ONOS instance 2	172.19.37.53
ONOS instance 3	172.19.37.54

TOPOLOGY DEVICE :-

Device's	Ip address
RT1	172.19.37.161
RT2	172.19.37.162

Step2:-
Import **Demo1.postman_collection.json** to postman.

Step3:-
Demo operation steps (on ONOS and Devices).

Start VPN(one time):
For US team's virtual machine:
```
cd /home/sdn/Downloads/ARRAY_VPN_CLIENT
./hacakathon_vpn_server_connect.py
```

Start ONOS(if onos is not started):
```
ssh onos@172.19.37.52
cd onos-1.10.0-SNAPSHOT/apache-karaf-3.0.8/bin
./karaf clean
ssh onos@172.19.37.53
cd onos-1.10.0-SNAPSHOT/apache-karaf-3.0.8/bin
./karaf clean
ssh onos@172.19.37.54
cd onos-1.10.0-SNAPSHOT/apache-karaf-3.0.8/bin
./karaf clean
```

Send below command to form cluster in your local onos
. tools/package/bin/onos-form-cluster 192.167.1.10 192.167.1.20 192.167.1.30

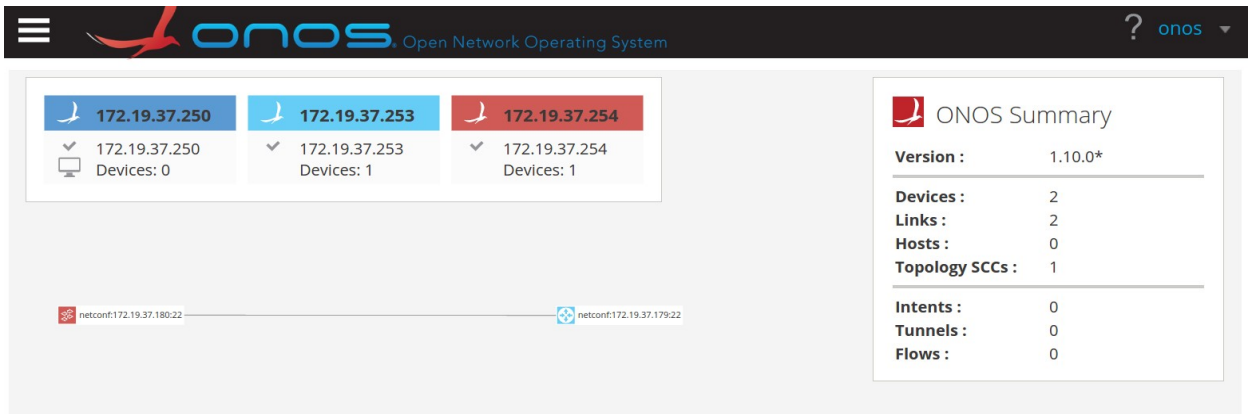
Wait for onos to start, check UI
```
check GUI http://172.19.37.52:8181/onos/ui
```

Step4:- Device's connection with onos

A) Device config post requests(**deviceconfig)** on to one of the onos instance via postman.
B) Devices can be seen on onos gui.

Step5:- Link discovery

A) Send link config post requests(**linkconfig)** on to one of the onos instance via postman.
B) Link between the devices can be observed on onos gui.



Note :- Below step are common for DEMO1 and DEMO2
only difference is json request should be sent from postman demo2-collection for demo2.

Steps 6:-
Pre-Observations:-
Ping should not work between devices.
start ping on RT1 using: ping -c 10000 -s 8000 -vpn-instance vrf1 10.1.1.2

Step 7:-
Send create via postman
A. Send instance create post requests(**RT1/RT2 create VPN)** on both devices.
B. Send interface create post requests(**RT1/RT2 create interface**) on both devices.
C. Send get request(**GetAll)** to verify post in store.
D. Check Routers' configuration:

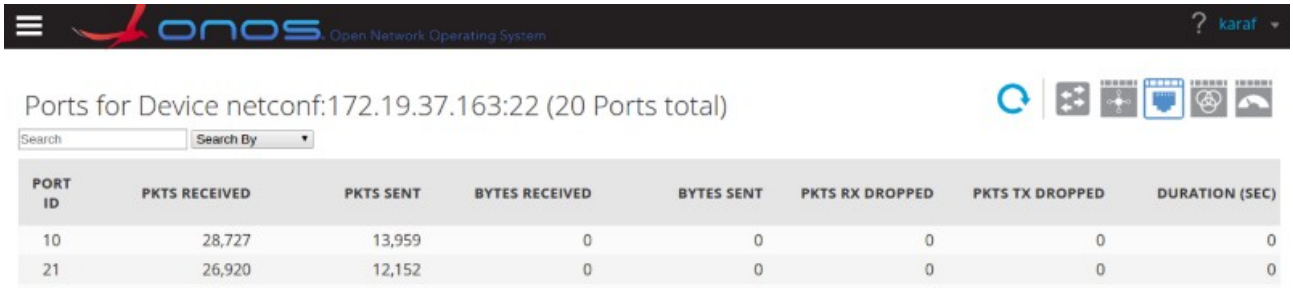
```
ssh telnet_admin@172.19.37.161
sys
display current-configuration
```

Vpn instance and interface should be configured on device as shown below:

```
ip vpn-instance vrf1
description vrfDescription
ipv4-family
route-distinguisher 100:1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
#
and
interface Ethernet3/0/4
undo shutdown
ip binding vpn-instance vrf1
ip address 10.1.1.1 255.255.255.0
#
```

same apply to Router 2 (172.19.37.162)

Observation points :-
A. Ping should work between devices.
B. Traffic flow can be observed between device ports.
C. To check packet stats on **link press 'a' three times.**



PORT ID	PKTS RECEIVED	PKTS SENT	BYTES RECEIVED	BYTES SENT	PKTS RX DROPPED	PKTS TX DROPPED	DURATION (SEC)
10	28,727	13,959	0	0	0	0	0
21	26,920	12,152	0	0	0	0	0

Step8:-
A. Send delete request**(deleteAll)** for both the devices.
B. Send get request**(GetAll)** to verify delete in store.

Limitations :
In step 7 instance create post should be sent always before interface create post.